11 | 负载均衡：节点负载差距这么大，为什么收到的流量还一样？

2020-03-13 何小锋

讲述：张浩
时长 11:21 大小 10.41M

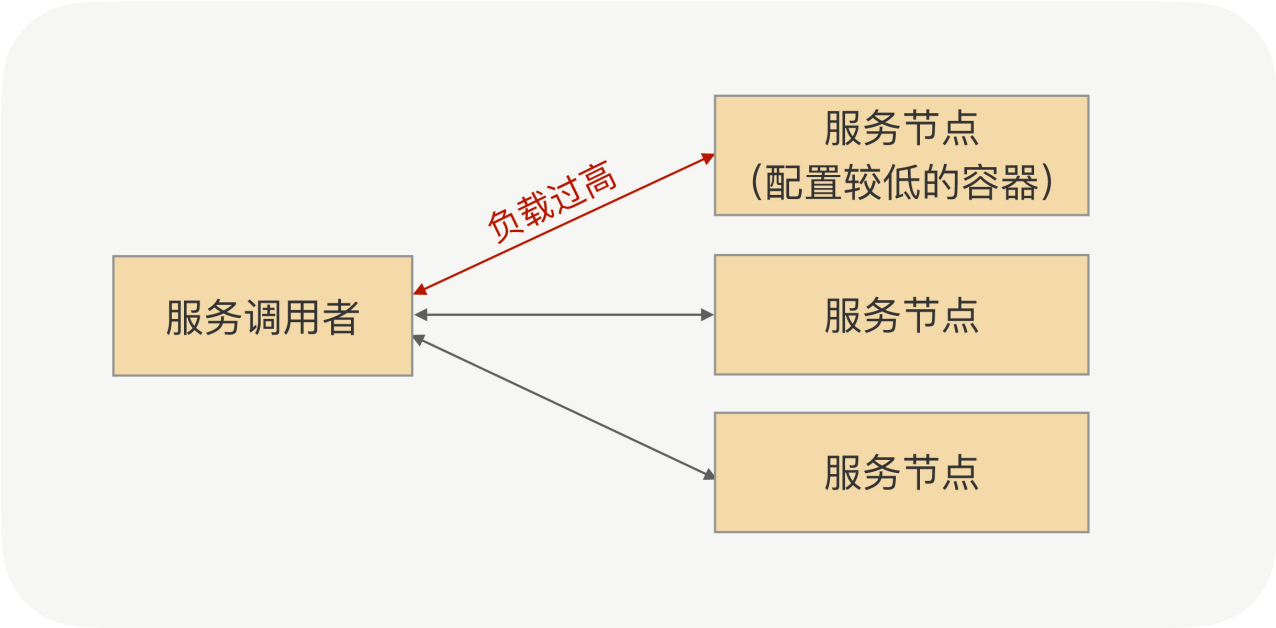
你好，我是何小锋。上一讲我讲解了“多场景的路由选择”，其核心就是“如何根据不同的场景控制选择合适的目标机器”。今天我们来聊一个新的话题，看看在 RPC 中如何实现负载均衡。

一个需求

在进入主题之前，我想先和你分享一个需求，这是我们公司的业务部门给我们提的。

他们反馈的问题是这样的：有一次碰上流量高峰，他们突然发现线上服务的可用率降低了，经过排查发现，是因为其中有几台机器比较旧了。当时最早申请的一批容器配置比较低，缩

容的时候留下了几台，当流量达到高峰时，这几台容器由于负载太高，就扛不住压力了。业务问我们有没有好的服务治理策略？



业务部门问题示意图

这个问题其实挺好解决的，我们当时给出的方案是：在治理平台上调低这几台机器的权重，这样的话，访问的流量自然就减少了。

但业务接着反馈了，说：当他们发现服务可用率降低的时候，业务请求已经受到影响了，这时再如此解决，需要时间啊，那这段时间里业务可能已经有损失了。紧接着他们就提出了需求，问：**RPC 框架有没有什么智能负载的机制？能否及时地自动控制服务节点接收到的访问量？**

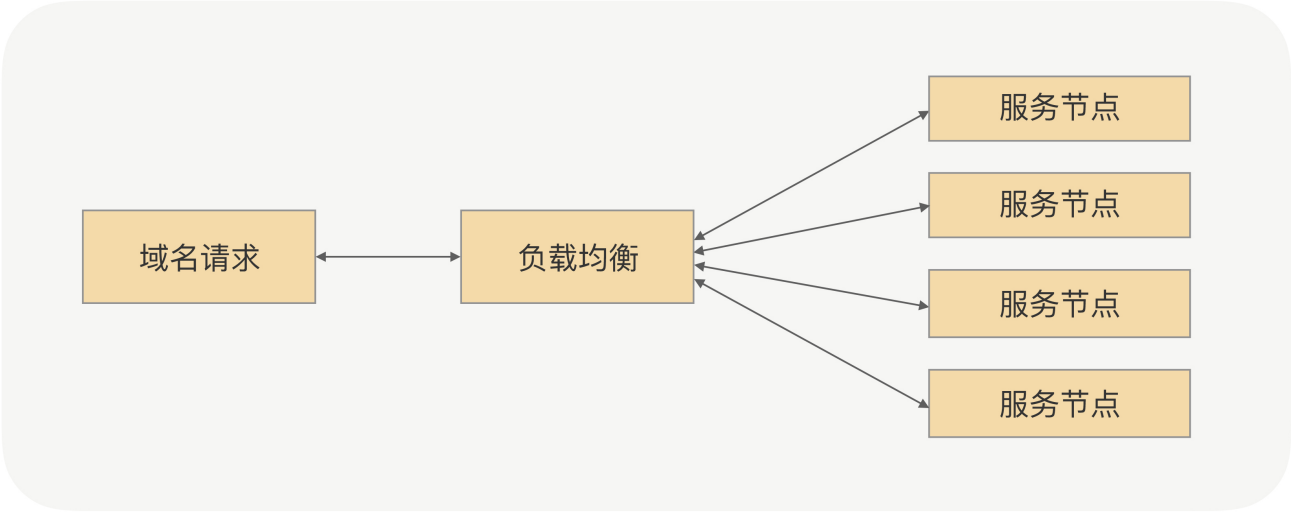
这个需求其实很合理，这也是一个比较普遍的问题。确实，虽说我们的服务治理平台能够动态地控制线上服务节点接收的访问量，但当业务方发现部分机器负载过高或者响应变慢的时候再去调整节点权重，真的很可能已经影响到线上服务的可用率了。

看到这儿，你有没有想到什么好的处理方案呢？接下来，我们就以这个问题为背景，一起看看 RPC 框架的负载均衡。

什么是负载均衡？

我先来简单地介绍下负载均衡。当我们的一个服务节点无法支撑现有的访问量时，我们会部署多个节点，组成一个集群，然后通过负载均衡，将请求分发给这个集群下的每个服务节

点，从而达到多个服务节点共同分担请求压力的目的。



负载均衡示意图

负载均衡主要分为软负载和硬负载，软负载就是在一台或多台服务器上安装负载均衡的软件，如 LVS、Nginx 等，硬负载就是通过硬件设备来实现的负载均衡，如 F5 服务器等。负载均衡的算法主要有随机法、轮询法、最小连接法等。

我刚才介绍的负载均衡主要还是应用在 Web 服务上，Web 服务的域名绑定负载均衡的地址，通过负载均衡将用户的请求分发到一个个后端服务上。

RPC 框架中的负载均衡

那 RPC 框架中的负载均衡是不是也是如此呢？和我上面讲的负载均衡，你觉得会有区别吗？

我们可以回想下[第 08 讲] 的开头：我讲到为什么不通过 DNS 来实现“服务发现”，之后我又讲解了为什么不采用添加负载均衡设备或者 TCP/IP 四层代理，域名绑定负载均衡设备的 IP 或者四层代理 IP 的方式。

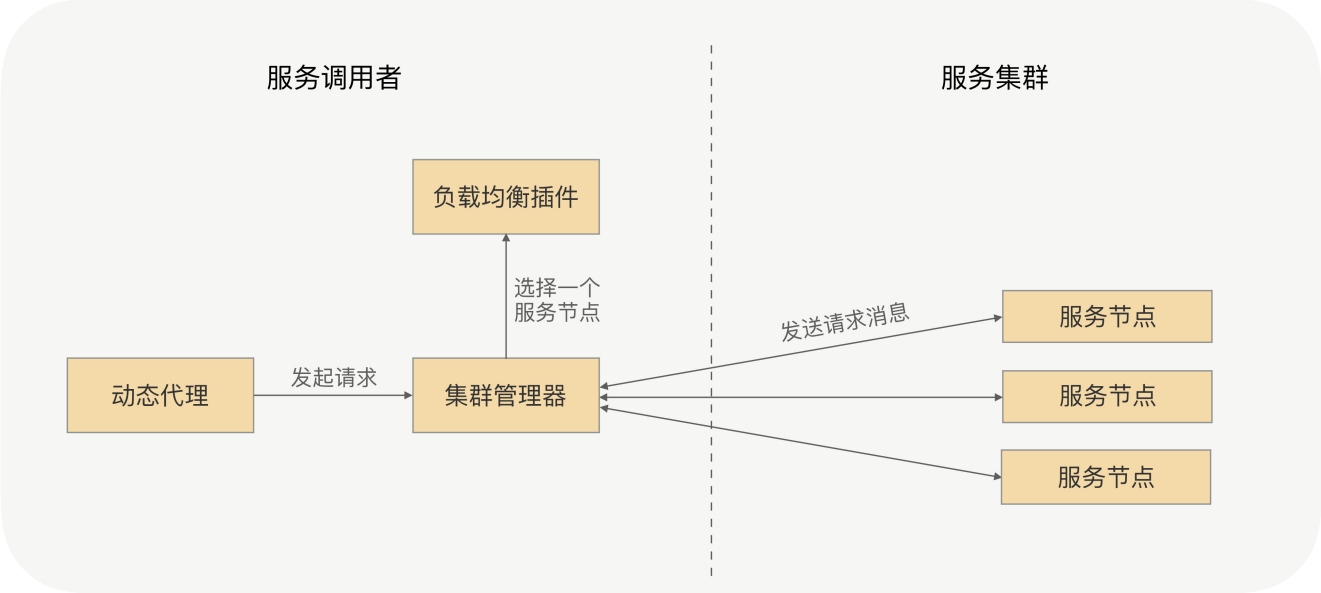
我的回答是这种方式会面临这样几个问题：

1. 搭建负载均衡设备或 TCP/IP 四层代理，需要额外成本；
2. 请求流量都经过负载均衡设备，多经过一次网络传输，会额外浪费一些性能；

3. 负载均衡添加节点和摘除节点，一般都要手动添加，当大批量扩容和下线时，会有大量的人工操作，“服务发现”在操作上是个问题；
4. 我们在服务治理的时候，针对不同接口服务、服务的不同分组，我们的负载均衡策略是需要可配的，如果大家都经过这一个负载均衡设备，就不容易根据不同的场景来配置不同的负载均衡策略了。

我相信看到这儿，你应该已经知道了 RPC 实现的负载均衡所采用的策略与传统的 Web 服务实现负载均衡所采用策略的不同之处了。

RPC 的负载均衡完全由 RPC 框架自身实现，RPC 的服务调用者会与“注册中心”下发的所有服务节点建立长连接，在每次发起 RPC 调用时，服务调用者都会通过配置的负载均衡插件，自主选择一个服务节点，发起 RPC 调用请求。



RPC框架负载均衡示意图

RPC 负载均衡策略一般包括随机权重、Hash、轮询。当然，这还是主要看 RPC 框架自身的实现。其中的随机权重策略应该是我们最常用的一种了，通过随机算法，我们基本可以保证每个节点接收到的请求流量是均匀的；同时我们还可以通过控制节点权重的方式，来进行流量控制。比如我们默认每个节点的权重都是 100，但当我们把其中的一个节点的权重设置成 50 时，它接收到的流量就是其他节点的 1/2。

这几种负载均衡算法的实现还是很简单的，网上资料也非常多，在这我就不过多介绍了。有什么问题，咱们可以在留言区交流。

由于负载均衡机制完全是由 RPC 框架自身实现的，所以它不再需要依赖任何负载均衡设备，自然也不会发生负载均衡设备的单点问题，服务调用方的负载均衡策略也完全可配，同时我们可以通过控制权重的方式，对负载均衡进行治理。

了解完 RPC 框架的负载均衡，现在我们就可以回到这讲最开头业务提的那个需求：有没有什么办法可以动态地、智能地控制线上服务节点所接收到的请求流量？

现在答案是不是就显而易见了，解决问题的关键就在于 RPC 框架的负载均衡上。对于这个问题，我们当时的方案就是，设计一种自适应的负载均衡策略。

如何设计自适应的负载均衡？

我刚才讲过，RPC 的负载均衡完全由 RPC 框架自身实现，服务调用者发起请求时，会通过配置的负载均衡插件，自主地选择服务节点。那是不是只要调用者知道每个服务节点处理请求的能力，再根据服务处理节点处理请求的能力来判断要打给它多少流量就可以了？当一个服务节点负载过高或响应过慢时，就少给它发送请求，反之则多给它发送请求。

这就有点像日常工作中的分配任务，要多考虑实际情况。当一位下属身体欠佳，就少给他些工作；若刚好另一位下属状态很好，手头工作又不是很多，就多分给他一点。

那服务调用者节点又该如何判定一个服务节点的处理能力呢？

这里我们可以采用一种打分的策略，服务调用者收集与之建立长连接的每个服务节点的指标数据，如服务节点的负载指标、CPU 核数、内存大小、请求处理的耗时指标（如请求平均耗时、TP99、TP999）、服务节点的状态指标（如正常、亚健康）。通过这些指标，计算出一个分数，比如总分 10 分，如果 CPU 负载达到 70%，就减它 3 分，当然了，减 3 分只是个类比，需要减多少分是需要一个计算策略的。

我们又该如果根据这些指标来打分呢？

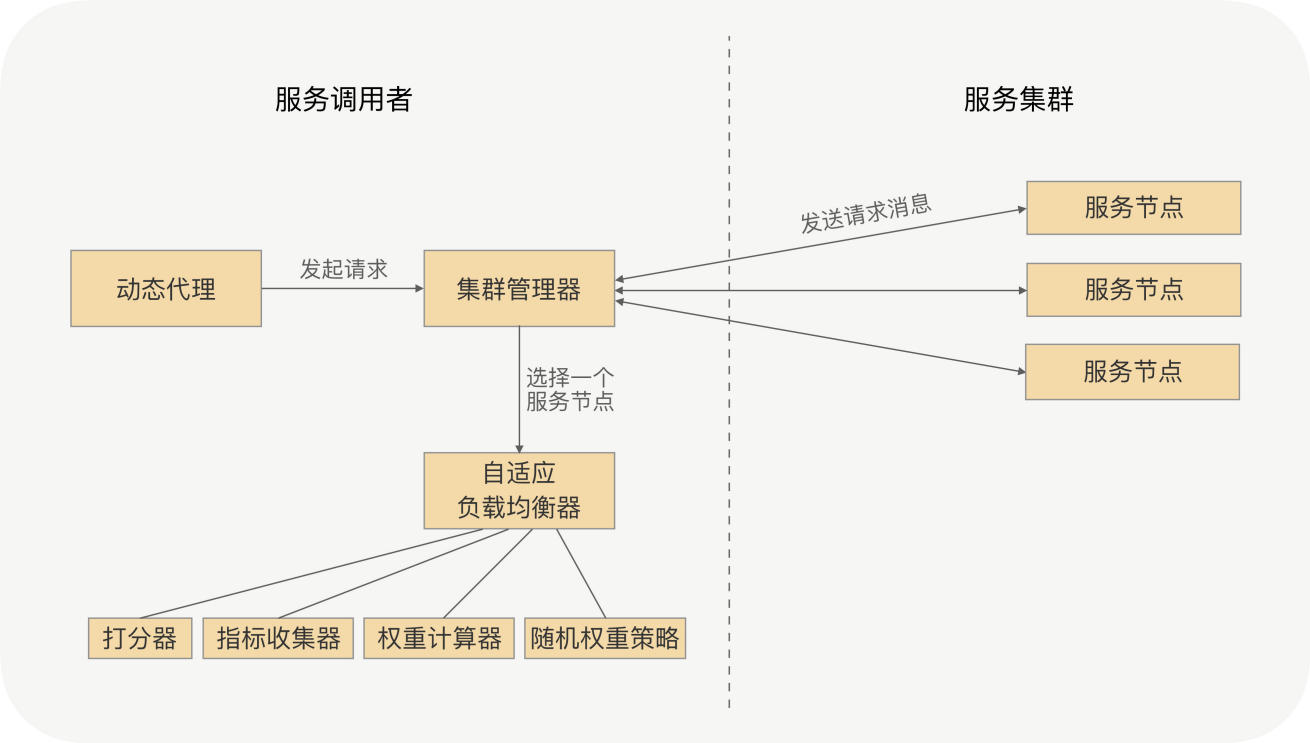
这就有点像公司对员工进行年终考核。假设我是老板，我要考核专业能力、沟通能力和工作态度，这三项的占比分别是 30%、30%、40%，我给一个员工的评分是 10、8、8，那他的综合分数就是这样计算的：10*30%+8*30%+8*40%=8.6 分。

给服务节点打分也一样，我们可以为每个指标都设置一个指标权重占比，然后再根据这些指标数据，计算分数。

服务调用者给每个服务节点都打完分之后，会发送请求，那这时候我们又该如何根据分数去控制给每个服务节点发送多少流量呢?

我们可以配合随机权重的负载均衡策略去控制，通过最终的指标分数修改服务节点最终的权重。例如给一个服务节点综合打分是 8 分（满分 10 分），服务节点的权重是 100，那么计算后最终权重就是 80（100*80%）。服务调用者发送请求时，会通过随机权重的策略来选择服务节点，那么这个节点接收到的流量就是其他正常节点的 80%（这里假设其他节点默认权重都是 100，且指标正常，打分为 10 分的情况）。

到这儿，一个自适应的负载均衡我们就完成了，整体的设计方案如下图所示：



RPC自适应负载均衡示意图

关键步骤我来解释下：

1. 添加服务指标收集器，并将其作为插件，默认有运行时状态指标收集器、请求耗时指标收集器。

2. 运行时状态指标收集器收集服务节点 CPU 核数、CPU 负载以及内存等指标，在服务调用者与服务提供者的心跳数据中获取。
3. 请求耗时指标收集器收集请求耗时数据，如平均耗时、TP99、TP999 等。
4. 可以配置开启哪些指标收集器，并设置这些参考指标的指标权重，再根据指标数据和指标权重来综合打分。
5. 通过服务节点的综合打分与节点的权重，最终计算出节点的最终权重，之后服务调用者会根据随机权重的策略，来选择服务节点。

总结

今天我们详细讲解了 RPC 框架的负载均衡，它与 Web 服务的负载均衡的不同之处在于：RPC 框架并不是依赖一个负载均衡设备或者负载均衡服务器来实现负载均衡的，而是由 RPC 框架本身实现的，服务调用者可以自主选择服务节点，发起服务调用。

这样的好处是，RPC 框架不再需要依赖专门的负载均衡设备，可以节约成本；还减少了与负载均衡设备间额外的网络传输，提升了传输效率；并且均衡策略可配，便于服务治理。

除此之外，我们今天的重点还涉及到“如何设计一个自适应的负载均衡”，通过它，我们可以就能根据服务调用者依赖的服务集群中每个节点的自身状态，智能地控制发送给每个服务节点的请求流量，防止因某个服务节点负载过高、请求处理过慢而影响到整个服务集群的可用率。

这个自适应负载均衡的实现方案，其实不只是应用于 RPC 框架中的负载均衡，它本身便是一个智能负载的解决方案，如果你在工作中需要设计一个智能的负载均衡服务，那么完全可以参考。

课后思考

你知道 RPC 框架中还有哪些负载均衡策略吗？它们的优缺点是什么？期待你能在留言区中分享实现方法，与我共同探讨。

也欢迎你把文章分享给你的朋友，邀请他加入学习。我们下节课再见！

更多课程推荐

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　10 | 路由策略：怎么让请求按照设定的规则发到不同的节点上?

下一篇　12 | 异常重试：在约定时间内安全可靠地重试

精选留言 (6)

写留言

楼下小黑哥
2020-03-13

以 Dubbo 为例，常用的负载均衡方法有：
1.基于权重随机算法
2.基于最少活跃调用数算法
3.基于 hash 一致性
4.基于加权轮询算法...

展开

作者回复: 课代表

4

安排
2020-03-13

是不是每个rpc调用方，也就是客户端都存在一个智能负载均衡？那就是每个rpc调用方都能掌握全局的负载信息了，要不然无法做负载均衡？其实还是对"负载均衡由rpc框架来做"不理解，这个rpc框架是每个rpc调用方都会有一份吗？

展开

作者回复: 负载均衡一般都需要内置在rpc里面，用于也可以进行扩展

1 1

FATMAN89
2020-03-17

请问老师再后续的课程中，是否会增加更多的代码实现，比如实现一个mini的rpc framework，当然老师对理论的讲解是非常精彩的

展开

小鱼儿
2020-03-15

老师，你好。 请问，有没有具体的框架可供实施呢？比如：consul，nginx，gRPC这些。

作者回复: 选择一些支持自定义扩展的

嬰
2020-03-14

谈谈GRPC的负载均衡策略，grpc官方并未提供服务发现注册的功能实现，但是为不同语言的gRPC代码实现提供了可扩展的命名解析和负载均衡接口。其基本原理是：服务启动后grpc客户端向命名服务器发出名称解析请求，名称会解析为一个或多个ip地址，每个ip地址会标识它是服务器地址还是负载均衡地址，以及标识要使用的那个客户端的负载均衡策略或服务配置。客户端实例化负载均衡策略，如果解析返回的地址是负载均衡器的地址...

展开

作者回复: 很棒!

唔多志

2020-03-13

有点疑惑，路由策略和负载均衡的结果都是选择一个合适的服务节点，那这两个有什么区别呢？

作者回复: 路由一般是规则设定，一般都是路由之后，负载再生效

1